

Comp512, Spring 2006

Lab 3 – Implementing an optimizer Due April 28th, 2006

The purpose of this lab is to give the student practical experience in choosing and implementing optimization passes.

Imagine that the marketing guy in your company has come to you and said, “We need you to implement code optimizations (whatever that means) into our existing compiler. I promised our clients that they could have it by the end of April and that it would run better than Nanosoft’s compiler.”

You only have time to implement three optimizations. What do you choose?

1 The lab

In this lab, you will select three optimizations and implement them in the ILOC research compiler. You should consider the optimizations covered in class, along with any of the optimizations covered in Chapter 10 of *Engineering a Compiler*. (Other sources are fine to consider, as well.) At least one of the optimizations must use ssa form.

How will you decide the optimizations? This is left as an open question: you should be guided by the taxonomy as well as lectures and discussion from class. The codes we will use to test the efficacy of your implementations are in *comp512/test* on owl.net.

The lab is structured as a competition between students. Twenty percent of your lab grade will be based on a comparison of the performance of your lab against the other students’ labs. The remainder of your lab grade will be based on correctness and efficiency – even if your compiler can’t beat Nanosoft’s, you’re going to lose *all* of your customers if the compiler produces garbage or runs insanely long!

You can run your three optimization passes in any order, in a sequence of up to ten passes (i.e., if your passes are labelled “A,” “B,” and “C,” you can specify that the compiler run “AABBCBAB”). You should create a runnable script called, “picosoft” that encodes the sequence you think is best. Put the script in the directory `you/512/lab3`. The script should act like any of the passes in the ILOC compiler: it should take either a file name or stdin, and it should put out the ILOC code to stdout. As a fail-safe, put the running instructions in a file called “README” in your `you/512/lab3` directory.

2 Advice

Start early.

Test your passes out of context – that is, run each pass independently against known (small) examples. Running an optimization pass in the midst of a collection of five to ten other passes can obscure problems and complicate your debugging efforts.

Think about command-line parameters that would be useful, both for the developer (in debugging) and for the end user (enabling or disabling features – these can have real impact on the performance of created code). The goal of command line options should be to provide the user with real control over the optimization and its application, rather than the classic `-O1`, `-O2`, `-O3`, . . . flags.